

Universidade Tiradentes  
Ciência da Computação

Caio Pierre Cardoso Braga  
Ramiro Goes Lisboa  
Gabriel Soares Silva Lima  
Luiz Felipe Prudente Borges  
Vinicius Vieira Lima

**PROCESSAMENTO DE IMAGENS DE COMPUTAÇÃO  
GRÁFICA E01**

**PROJETO UNIDADE 2**

Aracaju - SE  
2025

Caio Pierre Cardoso Braga  
Ramiro Goes Lisboa  
Gabriel Soares Silva Lima  
Luiz Felipe Prudente Borges  
Vinicius Vieira Lima

## **PROCESSAMENTO DE IMAGENS DE COMPUTAÇÃO GRÁFICA E01**

### **PROJETO UNIDADE 2**

ATIVIDADE sobre filtros de suavização e borda apresentado como requisito da avaliação da disciplina processamento de imagens de computação gráfica, ministrada pela Prof. Layse Santos Souza, no 8º semestre de 2025.

# Sumário

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>4</b>
<b>2</b>	<b>JUSTIFICATIVA . . . . .</b>	<b>5</b>
<b>3</b>	<b>OBJETIVOS . . . . .</b>	<b>6</b>
<b>3.1</b>	<b>OBJETIVO GERAL . . . . .</b>	<b>6</b>
<b>3.2</b>	<b>OBJETIVOS ESPECÍFICOS . . . . .</b>	<b>6</b>
<b>3.3</b>	<b>REQUISITOS FUNCIONAIS . . . . .</b>	<b>6</b>
<b>3.4</b>	<b>REQUISITOS NÃO-FUNCIONAIS . . . . .</b>	<b>6</b>
<b>4</b>	<b>METODOLOGIA . . . . .</b>	<b>8</b>
<b>4.1</b>	<b>Responsabilidades dos integrantes . . . . .</b>	<b>9</b>
<b>5</b>	<b>RESULTADOS E DISCUSSÕES . . . . .</b>	<b>10</b>
<b>5.1</b>	<b>Deteccção de bordas . . . . .</b>	<b>10</b>
5.1.1	Sobel . . . . .	11
5.1.2	Roberts . . . . .	12
5.1.3	Canny . . . . .	13
<b>5.2</b>	<b>Filtros de Blur . . . . .</b>	<b>14</b>
5.2.1	Gaussiano . . . . .	14
5.2.2	Bilateral . . . . .	15
5.2.3	Média . . . . .	16
5.2.4	Mediana . . . . .	17
<b>5.3</b>	<b>Filtros Personalizados . . . . .</b>	<b>17</b>
<b>6</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>20</b>
<b>7</b>	<b>REFERÊNCIAS . . . . .</b>	<b>21</b>
<b>8</b>	<b>ANEXOS . . . . .</b>	<b>22</b>

# 1 INTRODUÇÃO

O avanço das tecnologias de visão computacional transformou a análise de imagens em uma ferramenta essencial para diversas áreas, como segurança pública, mobilidade urbana, saúde, automação industrial e sistemas inteligentes. A crescente capacidade de capturar, armazenar e processar grandes volumes de dados visuais impulsionou o desenvolvimento de soluções capazes de interpretar cenas com rapidez e precisão. No campo específico da segurança, esses recursos se tornaram fundamentais para o monitoramento de ambientes, reconhecimento de padrões e identificação de eventos relevantes, atuando como suporte estratégico para tomadas de decisão em tempo real.

Nesse cenário, técnicas de pré-processamento, como suavização e detecção de bordas, desempenham um papel decisivo no desempenho de sistemas de visão computacional. A suavização reduz ruídos que podem comprometer a interpretação da imagem, enquanto a detecção de bordas permite evidenciar transições abruptas de intensidade, realçando elementos estruturais importantes para etapas posteriores de análise. A combinação dessas técnicas constitui a base para tarefas mais complexas, como segmentação, reconhecimento de objetos e rastreamento.

O presente trabalho tem como objetivo o desenvolvimento de um sistema de processamento de imagens que aplica filtros clássicos de suavização, como Média e Gaussiano, e métodos consolidados de detecção de bordas, como Sobel e Canny. Cada técnica é explorada de forma a demonstrar seu funcionamento, suas vantagens e suas limitações, possibilitando compreender como diferentes abordagens influenciam o resultado final. Além disso, o projeto inclui a análise de metadados associados às imagens, ampliando a compreensão do contexto em que elas foram capturadas.

A proposta busca não apenas demonstrar o funcionamento isolado de cada método, mas também apresentar sua relevância prática em cenários reais, especialmente em aplicações de segurança pública e monitoramento urbano. Ao integrar processamento de imagens e análise contextual, o sistema desenvolvido oferece uma visão clara sobre como técnicas fundamentais da visão computacional podem ser aplicadas na construção de soluções mais robustas, eficientes e confiáveis.

## 2 JUSTIFICATIVA

A interpretação visual de elementos lineares em imagens é um procedimento subjetivo, o que torna necessária a automatização da extração dessas informações por meio de métodos computacionais. Já existem iniciativas nesse sentido, como o uso de operadores para realce e detecção de bordas.

Dentro do que diz respeito ao propósito geral do nosso projeto, objetivamos contribuir para com o monitoramento, segurança, fiscalização e auxílio de órgãos de segurança com a utilização de filtros de suavização e detecção de bordas. Esses filtros já são amplamente utilizados no processamento de imagens de sensoriamento remoto e visão computacional aplicados a ambientes urbanos, com o objetivo de extrair informações úteis para o planejamento e monitoramento das cidades. Dentre os exemplos que podemos citar quanto a detecção de bordas utilizadas em ambientes urbanos estão:

- Usada para identificar e delinear automaticamente os contornos de edifícios, estradas, rios e outras características geográficas a partir de imagens de satélite ou aéreas.
- Auxilia na criação e atualização de mapas urbanos precisos e na análise da morfologia urbana.
- Sistemas de vigilância por vídeo: a detecção de bordas ajuda a identificar e rastrear veículos e pedestres, segmentando-os do fundo da imagem.
- Ajuda a delimitar diferentes zonas de uso do solo residencial, comercial, industrial, áreas verdes em imagens de satélite, permitindo que planejadores urbanos avaliem a expansão da cidade e a conformidade com os planos diretores.
- Em câmeras de segurança, os algoritmos de detecção de bordas são usados para identificar pessoas ou objetos suspeitos com base em sua forma e movimento, acionando alertas automáticos.
- Usada para delimitar áreas de desmatamento, expansão de áreas agrícolas e mudanças na cobertura vegetal em áreas periurbanas, auxiliando na gestão ambiental e no controle de poluição.
- Os dados de Detecção e Alcance de Luz podem ser suavizados para criar modelos digitais de terreno mais uniformes e precisos, essenciais para o planejamento de infraestrutura e gestão de inundações.
- A redução de ruído facilita a segmentação, ajudando a agrupar áreas com características semelhantes, como zonas residenciais, parques industriais e corpos d'água.

Com isso em mente, nossa justificativa gira em torno da contribuição para o sistema de segurança e monitoramento de cidades onde a interpretação de elementos está diretamente presente.

## 3 OBJETIVOS

### 3.1 OBJETIVO GERAL

Desenvolver um sistema capaz de realizar processamento de imagens com técnicas de suavização, detecção de bordas e comparação entre versões originais e processadas, assim como apresentar metadados das imagens originais e com filtros.

### 3.2 OBJETIVOS ESPECÍFICOS

1. Implementar filtros de suavização (Média, Mediana, Gaussiano, Bilateral);
2. Aplicar operadores de detecção de bordas (Sobel, Roberts, Canny);
3. Permitir a exportação da imagem em diferentes formatos;
4. Exibir metadados da imagem antes e após o processamento.

### 3.3 REQUISITOS FUNCIONAIS

- Comparativo da imagem original e modificada assim como o histograma;
- Fácil anexo de imagem;
- Deve aceitar imagens apenas nos formatos .png ou .jpg ou .jpeg;
- Aplicação de filtros de suavização;
- Ajuste de parâmetros de filtro;
- Salvar a imagem em outro formato;
- O usuário deve ser capaz de aplicar uma sequência de filtros na mesma imagem;
- O sistema deve exibir metadados da imagem, como resolução (largura x altura), tamanho do arquivo, e tipo de arquivo, antes e após o processamento;
- O sistema deve permitir o anexo de imagens de forma simples;
- O sistema deve permitir a aplicação de filtros de detecção de bordas.

### 3.4 REQUISITOS NÃO-FUNCIONAIS

- A qualidade da detecção de bordas será avaliada pela continuidade e precisão dos contornos principais da imagem, utilizando métricas como a porcentagem de pixels conectados e o desvio padrão de regiões homogêneas após filtragem;
- O sistema deve ser capaz de processar imagens de alta resolução sem degradação significativa no desempenho;

- O sistema deve processar imagens sem falhas ou corrupção de dados;
- Em caso de entrada de dados inválida (ex: arquivo que não é uma imagem), o sistema deve tratar a exceção de forma robusta e não falhar;
- O sistema deve fornecer mensagens de erro claras e informativas em caso de falha no carregamento ou processamento da imagem;
- O sistema deve permitir que o usuário escolha o formato de saída (JPG, PNG, JPEG);
- O sistema deve apresentar tempo de resposta aceitável.

## 4 METODOLOGIA

A metodologia adotada para o desenvolvimento do sistema foi dividida em etapas:

- Análise e planejamento inicial;
- Modelagem do sistema e definição de funções;
- Implementação dos filtros e operadores;
- Testes e validação do desempenho.

### Estruturação do fluxo de processamento

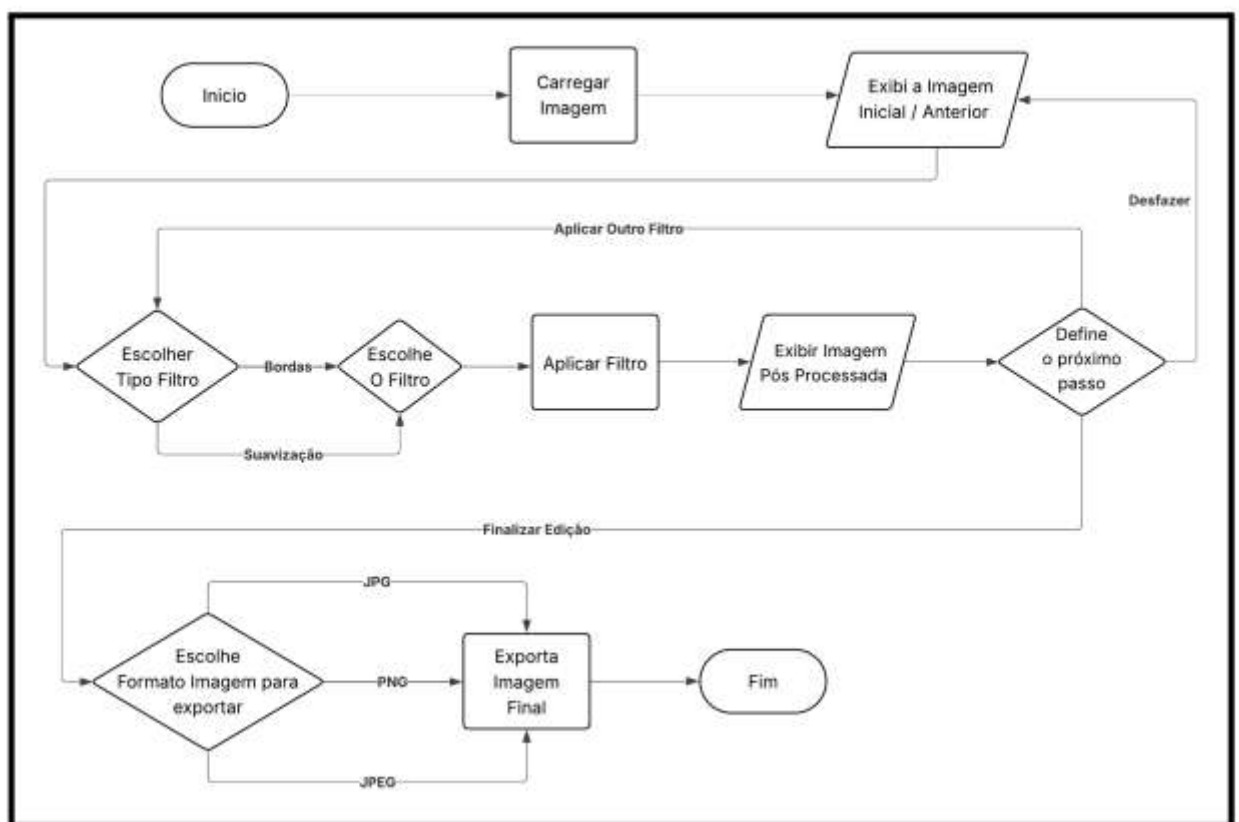


Figura 1 – Estrutura de Fluxo de Processamento

As bibliotecas OpenCV e Scikit-Image foram consideradas como ferramentas principais para o processamento digital de imagens.

Sendo utilizados os seguintes filtros e operadores:

- Filtro da Média: `cv2.blur(src, ksize)`
  - Aplica o filtro da média para suavizar a imagem, borrando bordas.
- Filtro Mediana: `cv2.medianBlur(src, ksize)`
  - Remove o ruído "sal e pimenta" de forma eficaz, preservando as bordas.
- Remove o ruído "sal e pimenta" de forma eficaz, preservando as bordas.
  - Suaviza a imagem com um borrão mais natural.

- Filtro Bilateral: `cv2.bilateralFilter(src, d, sigmaColor, sigmaSpace)`
  - Suaviza o ruído, mas preserva as bordas, ideal para detalhes.
- Operador Sobel: `cv2.Sobel(src, ddepth, dx, dy, ksize)`
  - Detecta bordas, calculando o gradiente da imagem em direções específicas.
- Operador Roberts: `skimage.filters.roberts(src)`
  - Usa um kernel 2x2 para detectar bordas de forma simples e rápida.
- Operador Canny: `cv2.Canny(src, threshold1, threshold2)`
  - Encontra bordas finas e contínuas, sendo o padrão para alta qualidade.

## 4.1 Responsabilidades dos integrantes

Vinicius Vieira:

- Coordenação geral do projeto e integração dos módulos;
- Implementação dos filtros de suavização (Média e Mediana);
- Documentação técnica e relatório final.

Luiz Felipe Prudente:

- Modelagem do sistema e definição da arquitetura;
- Implementação dos filtros de suavização (Gaussiano e Bilateral);
- Testes de desempenho com imagens de alta resolução;

Caio Pierre:

- Desenvolvimento da interface do usuário;
- Implementação do sistema de carregamento e exportação de imagens;
- Tratamento de exceções e validação de entrada de dados;

Ramiro Goes:

- Implementação dos operadores de detecção de bordas (Sobel e Roberts);
- Sistema de exibição de metadados das imagens;
- Testes de qualidade de detecção de bordas;

Gabriel Soares:

- Implementação do operador Canny;
- Desenvolvimento do sistema de comparação (imagem original vs. processada e histogramas);
- Validação final e testes de integração;

## 5 RESULTADOS E DISCUSSÕES

No início do desenvolvimento do projeto, foi-se realizado a utilização base dos códigos trabalhados em sala de aula e com os conhecimentos obtidos com a progressão do assunto implementamos individualmente os códigos fazendo o melhoramento de cada filtro individualmente. O resultado geral foi a criação de uma API que possibilita a utilização dos respectivos filtros de suavização e detecção de borda com a escolha padrão de definição, bem como filtros personalizados para definição pelo usuário.

A utilização decorre da seguinte forma, ao ter acesso aos filtros, pode-se escolher os filtros especificismos a serem utilizados. Ao fazê-lo, utilizando a opção de "try out", é possível fazer o upload de um arquivo de imagem. A partir daí, o programa gera um arquivo baixáveis, qual é possível fazer o download qual apresentará um arquivo compactado contendo tanto a imagem original, quanto a imagem modificada pelo programa. O sistema também oferece as opções de gerarem um arquivo em Json com a imagem alterada qual pode ser acessada utilizando a Base64 to image, onde podemos fazer sua visualização.

Dentre todas as soluções desenvolvidas seguem os resultados de cada campo específico com seus respectivos testes.

### 5.1 Detecção de bordas

Na detecção de bordas temos a utilização para identificar descontinuidades de intensidade que indicam os limites entre objetos ou regiões distintas. Diversos operadores podem ser aplicados, cada um com características específicas: o filtro Sobel destaca bordas suaves ao calcular gradientes horizontais e verticais, sendo pouco sensível ao ruído; o filtro Roberts utiliza uma máscara simples de  $2 \times 2$ , proporcionando rapidez na detecção, porém com maior sensibilidade a ruídos; e o filtro Canny é considerado o mais robusto, pois combina suavização, cálculo de gradientes e limiares adaptativos para obter bordas finas, contínuas e com menor incidência de falsos positivos. A escolha do método mais adequado depende do tipo de imagem e do nível de detalhe desejado na análise.

### 5.1.1 Sobel



(a) Imagem Original.



(b) Filtro nível 1.



(c) Filtro nível 2.



(d) Filtro nível 3.

Figura 2 – Filtro de Sobel.

O operador de Sobel é um método clássico de detecção de bordas que utiliza máscaras convolucionais para estimar o gradiente da intensidade da imagem nas direções horizontal e vertical. Na imagem original, observam-se regiões com variações nítidas de contraste, elementos ideais para aplicação do filtro. No nível 1, o Sobel começa a realçar bordas mais suaves, permitindo identificar os contornos principais sem gerar excesso de ruído. Já no nível 2, há um aumento na intensidade do gradiente, tornando as bordas mais espessas e mais brilhantes, o que evidencia melhor as transições bruscas de intensidade. No nível 3, o resultado apresenta bordas ainda mais destacadas e, em alguns casos, um aumento na sensibilidade a pequenas variações da imagem.

Teoricamente, o Sobel produz bordas contínuas e com certo grau de suavização devido ao seu kernel que incorpora uma ponderação central, reduzindo ruídos em comparação ao operador simples. Os resultados obtidos estão em conformidade com esse comportamento esperado: à medida que o nível do filtro aumenta, as bordas tornam-se mais claras e definidas, sem excesso de fragmentação. Assim, confirma-se que o operador Sobel funcionou corretamente e produziu a detecção de bordas conforme previsto pela literatura.

### 5.1.2 Roberts



(a) Imagem Original.



(b) Filtro nível 1.



(c) Filtro nível 2.



(d) Filtro nível 3.

Figura 3 – Filtro de Roberts.

O operador de Roberts utiliza máscaras  $2 \times 2$  para detectar variações diagonais de intensidade. Por utilizar kernels extremamente pequenos, é um método rápido, porém mais sensível a ruídos. Na imagem original, observam-se áreas com texturas e detalhes finos, elementos que tendem a ser ressaltados. No nível 1, as bordas aparecem de forma relativamente suave, porém é possível perceber maior granularidade. No nível 2, essa característica intensifica-se, e detalhes mínimos começam ampliando o ruído visual. No nível 3, o operador evidencia praticamente qualquer variação de intensidade, produzindo um mapa de bordas mais denso e fragmentado.

O Roberts é ideal para imagens com pouco ruído ou quando se deseja uma detecção rápida de bordas diagonais. Como a imagem utilizada contém regiões de contraste variado, o comportamento observado aumento progressivo de ruído e fragmentação está de acordo com o esperado. O filtro funcionou corretamente, porém reforça a limitação natural do método em ambientes ruidosos, justificando sua menor utilização em aplicações que exigem resultados mais limpos.

### 5.1.3 Canny



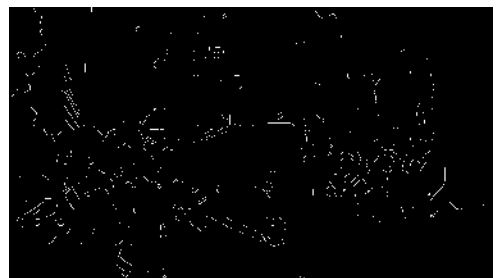
(a) Imagem Original.



(b) Filtro nível 1.



(c) Filtro nível 2.



(d) Filtro nível 3.

Figura 4 – Filtro de Canny.

Canny é considerado o método mais robusto de detecção de bordas, pois combina suavização Gaussiana, cálculo do gradiente, supressão de não máximos e histerese de limiares. Na imagem original, é possível identificar regiões de forte contraste que servem como referência para o operador. No nível 1, o Canny produz bordas finas, contínuas e bem conectadas, destacando apenas as transições mais significativas. No nível 2, os limiares ajustados tornam o filtro mais sensível, aumentando o número de bordas detectadas. No nível 3, o operador evidencia praticamente todos os gradientes relevantes, porém preserva a característica de bordas bem definidas e com baixa incidência de ruído.

Comparado à teoria, o Canny apresenta resultados semelhantes aos esperados: bordas nítidas, conectadas e com controle refinado da sensibilidade. Mesmo no nível mais alto, o algoritmo mantém a supressão de não máximos, impedindo que o mapa final fique saturado ou fragmentado. Dessa forma, o filtro funcionou perfeitamente, reafirmando sua posição como o detector mais confiável entre os métodos analisados.

## 5.2 Filtros de Blur

### 5.2.1 Gaussiano



(a) Imagem Original.



(b) Filtro nível 1.



(c) Filtro nível 2.



(d) Filtro nível 3.

Figura 5 – Filtro Gaussiano.

O filtro Gaussiano aplica uma suavização baseada na distribuição normal, reduzindo ruídos ao mesmo tempo em que preserva parcialmente as bordas. A imagem original

apresenta algumas variações de textura que podem ser suavizadas com eficiência por esse método. No nível 1, observa-se um leve borramento que diminui pequenas irregularidades sem comprometer a estrutura geral. No nível 2, o efeito se intensifica, suavizando regiões maiores e deixando os contornos mais difusos. No nível 3, o borramento é bastante perceptível, e detalhes menores desaparecem, produzindo um aspecto mais homogêneo.

Segundo a teoria, o Gaussiano produz suavização progressiva conforme o aumento do desvio padrão e do tamanho do kernel. Isso corresponde exatamente ao observado: níveis mais altos resultam em perda de detalhes, redução de ruído e transições mais suaves. O filtro operou corretamente, demonstrando coerência entre o comportamento empírico e o teórico.

### 5.2.2 Bilateral



(a) Imagem Original.



(b) Filtro nível 1.



(c) Filtro nível 2.



(d) Filtro nível 3.

Figura 6 – Filtro Bilateral.

O filtro Bilateral é conhecido por suavizar a imagem preservando as bordas, pois combina distância espacial e diferença de intensidade para limitar a influência de vizinhos distantes ou muito diferentes. Na imagem original, vários detalhes e bordas finas poderiam ser distorcidos por filtros tradicionais, mas são adequados para análise pelo Bilateral. No nível 1, percebe-se uma leve suavização do ruído sem perda perceptível de contornos. No nível 2, o borramento aumenta, mas as bordas principais permanecem bem delimitadas.

No nível 3, a imagem torna-se visivelmente mais suave, porém ainda preserva de forma eficiente os limites entre objetos.

De acordo com a teoria, o Bilateral deve manter bordas nítidas enquanto reduz ruídos algo que nem o Gaussiano nem a Média conseguem fazer com a mesma eficiência. Isso é confirmado pelos resultados: mesmo no nível mais elevado, não ocorre perda abrupta de estrutura. O filtro evidenciou seu funcionamento esperado e mostrou-se adequado para aplicações em que se deseja suavização sem comprometer contornos.

### 5.2.3 Média



(a) Imagem Original.



(b) Filtro nível 1.



(c) Filtro nível 2.



(d) Filtro nível 3.

Figura 7 – Filtro Média.

O filtro da Média calcula a média dos valores dos pixels em uma janela, resultando em uma suavização simples, porém propensa a borrar bordas. Na imagem original, vê-se diversas transições nítidas que são naturalmente afetadas por esse tipo de filtro. No nível 1, a suavização é leve, e a perda de detalhes é discreta. No nível 2, a imagem torna-se mais homogênea, com redução significativa de ruídos, mas com evidente borramento das bordas. No nível 3, o efeito intensifica-se, criando regiões visivelmente lisas e eliminando detalhes estruturais importantes.

Teoricamente, o filtro da Média é o método menos sofisticado, e seu comportamento condiz exatamente com o observado: borramento progressivo, perda de nitidez e redução do contraste local. O filtro funcionou conforme previsto, mas evidencia suas limitações em

preservar detalhes reforçando por que filtros mais avançados, como Gaussiano e Bilateral, são geralmente preferidos.

#### 5.2.4 Mediana



(a) Imagem Original.



(b) Filtro nível 1.



(c) Filtro nível 2.



(d) Filtro nível 3.

Figura 8 – Filtro Média.

O filtro da Mediana substitui cada pixel pela mediana de seus vizinhos, sendo extremamente eficiente na remoção de ruído impulsivo (“sal e pimenta”). Na imagem original, há regiões em que esse tipo de ruído pode ser percebido em menor ou maior grau. No nível 1, a suavização é eficiente e preserva bem as bordas, característica típica do método. No nível 2, o efeito se intensifica, reduzindo ainda mais ruídos, mas mantendo a integridade estrutural da imagem. No nível 3, observa-se uma suavização mais forte, mas sem borramento excessivo — diferentemente do filtro da Média.

Teoricamente, a Mediana preserva bordas muito melhor que filtros lineares, pois não depende da soma ou média de intensidades. Isso é confirmado pelos resultados obtidos: mesmo no nível máximo, as bordas continuam definidas, e o ruído é reduzido de maneira eficaz. O desempenho está em total conformidade com o comportamento esperado do filtro, comprovando sua utilidade em imagens com variações abruptas ou ruídos impulsivos.

### 5.3 Filtros Personalizados

Os filtros personalizados permitem ao usuário definir manualmente parâmetros de diferentes filtros, oferecendo um controle mais preciso sobre o comportamento do filtro

aplicado. Essa flexibilidade possibilita criar resultados adaptados ao contexto da imagem e à necessidade específica de suavização ou realce de bordas.

A seguir, é apresentada a análise dos resultados obtidos com dois filtros personalizados implementados: um Blur Gaussiano com parâmetros definidos pelo usuário e um Canny Personalizado com valores específicos de limiar e detecção de gradiente.



(a) Imagem Original.



(b) Filtro Personalizado.

Figura 9 – Filtro Gaussiano Personalizado.

Na imagem filtrada pelo Gaussiano com parâmetros Largura do Kernel = 33, Altura do Kernel = 7 e Sigma = 5, é possível perceber imediatamente um desfoque muito mais intenso e direcionado do que nos níveis padrão mostrados na seção 5.2.1. Isso ocorre porque o tamanho do kernel define a área da vizinhança considerada no cálculo da média ponderada do Gaussiano.

- Kernel  $33 \times 7$

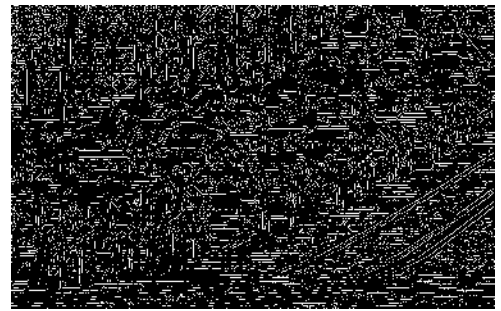
- Um kernel alongado dessa forma produz uma suavização assimétrica. Ele suaviza muito mais em uma direção do que em outra, o que faz com que a imagem apresente um borramento predominante horizontal (33 px) comparado ao vertical (7 px).
- Esse comportamento está alinhado com a teoria: quanto maior o kernel, maior o grau de suavização e perda de detalhe.

- Sigma = 5

- Um sigma relativamente alto aumenta o peso das médias e expande o efeito do borramento, fazendo com que até variações moderadas de intensidade dentro da área do kernel sejam suavizadas.
- Em comparação aos filtros Gaussianos dos níveis 1, 2 e 3 apresentados anteriormente, este filtro personalizado gera um efeito significativamente mais forte, ultrapassando até mesmo o nível 3 mostrado na Figura 5. Isso confirma que o filtro funcionou conforme o esperado, pois a teoria estabelece que sigma controla a “largura” da curva de suavização.



(a) Imagem Original.



(b) Filtro Personalizado.

Figura 10 – Filtro de Canny Personalizado.

Na aplicação do Canny personalizado foram utilizados os parâmetros Primeira Limiar = 1, Segunda limiar = 235, Tamanho da Abertura Sobel = 5 e não foi aplicado o blur gaussiano antes da detecção. Esses valores produzem um resultado visual distinto das versões apresentadas nos níveis da Figura 4.

- Primeira Limiar = 1
  - Um limiar inferior extremamente baixo tende a marcar praticamente qualquer variação mínima de intensidade como borda candidata. Isso aumenta o número de pixels inicialmente considerados como bordas fracas.
- Segunda Limiar = 235
  - Um limiar superior muito alto restringe fortemente quais dessas bordas fracas serão confirmadas como bordas reais. Essa combinação extrema gera um comportamento em que muitas bordas são detectadas, mas poucas sobrevivem ao processo de histerese, resultando em um desenho com varias bordas.
  - Esse parâmetro aumenta a sensibilidade do cálculo do gradiente usando Sobel. Kernels maiores captam variações mais suaves, o que pode identificar bordas mais largas, mas torna o detector mais suscetível a ruídos quando não existe suavização prévia.
  - Como o Canny depende fortemente da suavização, desativar o blur faz com que ruídos da imagem original interfiram no cálculo do gradiente.

## 6 CONSIDERAÇÕES FINAIS

O desenvolvimento do projeto possibilitou a aplicação prática dos principais conceitos de processamento digital de imagens, especialmente no uso de filtros de suavização e detecção de bordas. A implementação das técnicas estudadas como os filtros de Média, Mediana, Gaussiano e Bilateral e seus operadores possibilitou compreender de forma o impacto dos métodos sobre a qualidade visual e a precisão dos resultados obtidos.

A construção da API tornou possível integrar todas as ferramentas desenvolvidas, possibilitando que o usuário aplique filtros individualmente ou em sequência, visualize diferenças entre versões processadas e originais e acesse metadados relacionados à imagem. O comportamento obtido durante os testes demonstrou que pequenas alterações nos parâmetros podem gerar resultados significativamente distintos, o que reforça a importância da escolha criteriosa de valores como o tamanho dos kernels, o sigma do Gaussiano e os limiares utilizados no Canny.

A inclusão dos filtros personalizados também evidenciou o potencial de flexibilidade do sistema, permitindo ao usuário adaptar o processamento conforme o contexto da imagem ou o objetivo desejado — seja enfatizar detalhes, suavizar ruídos ou realçar bordas fortes. Os resultados confirmaram que a implementação segue corretamente a teoria apresentada na literatura, validando tanto a precisão dos filtros quanto a robustez do ambiente construído.

A integração entre os diferentes módulos do sistema demonstrou a importância do pré-processamento na redução de ruídos e na preparação das imagens para análises mais complexas. O uso combinado de filtros e operadores evidenciou como pequenas variações nos parâmetros podem alterar significativamente o resultado final, reforçando a necessidade de ajustes criteriosos conforme o contexto da aplicação.

Além disso, o projeto proporcionou uma experiência valiosa de trabalho em equipe, com divisão de tarefas clara e colaboração efetiva entre os integrantes. A utilização de bibliotecas amplamente empregadas na área, como OpenCV e Scikit-Image, contribuiu para o aprendizado de ferramentas práticas e consolidadas no mercado.

Por fim, o sistema desenvolvido alcançou os objetivos propostos, apresentando resultados satisfatórios em termos de desempenho, precisão e funcionalidade. O estudo serviu como base para futuras melhorias, como a inclusão de novos algoritmos de segmentação e reconhecimento de padrões, ampliando o potencial de aplicação em áreas como segurança pública, monitoramento urbano e análise de imagens em tempo real.

## 7 REFERÊNCIAS

GONZALEZ, Rafael C.; WOODS, Richard E. Digital Image Processing. 4. ed. Pearson, 2018.

JAIN, Anil K. Fundamentals of Digital Image Processing. Prentice-Hall, 1989.

CANNY, John. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986.

OpenCV Documentation. Image Filtering – Gaussian Blur, Bilateral Filter, Median Blur, Convolution. Disponível em: <https://docs.opencv.org/>

OpenCV Documentation. Canny Edge Detector. Disponível em: <https://docs.opencv.org/>

SCIKIT-IMAGE Documentation. Filters and Edge Detection (Roberts, Sobel, Prewitt). Disponível em: <https://scikit-image.org/>

SZE, Vivian. EECS 6.341 – Digital Signal Processing – Massachusetts Institute of Technology (MIT).

BUJALESKI, Filip et al. Image Smoothing Filters Review and Comparative Analysis. Journal of Information Technology and Applications, 2018.

TOMASI, Carlo; MANDUCHI, Roberto. Bilateral Filtering for Gray and Color Images. In: IEEE ICCV, 1998.

## 8 ANEXOS

### Estruturação do fluxo de processamento

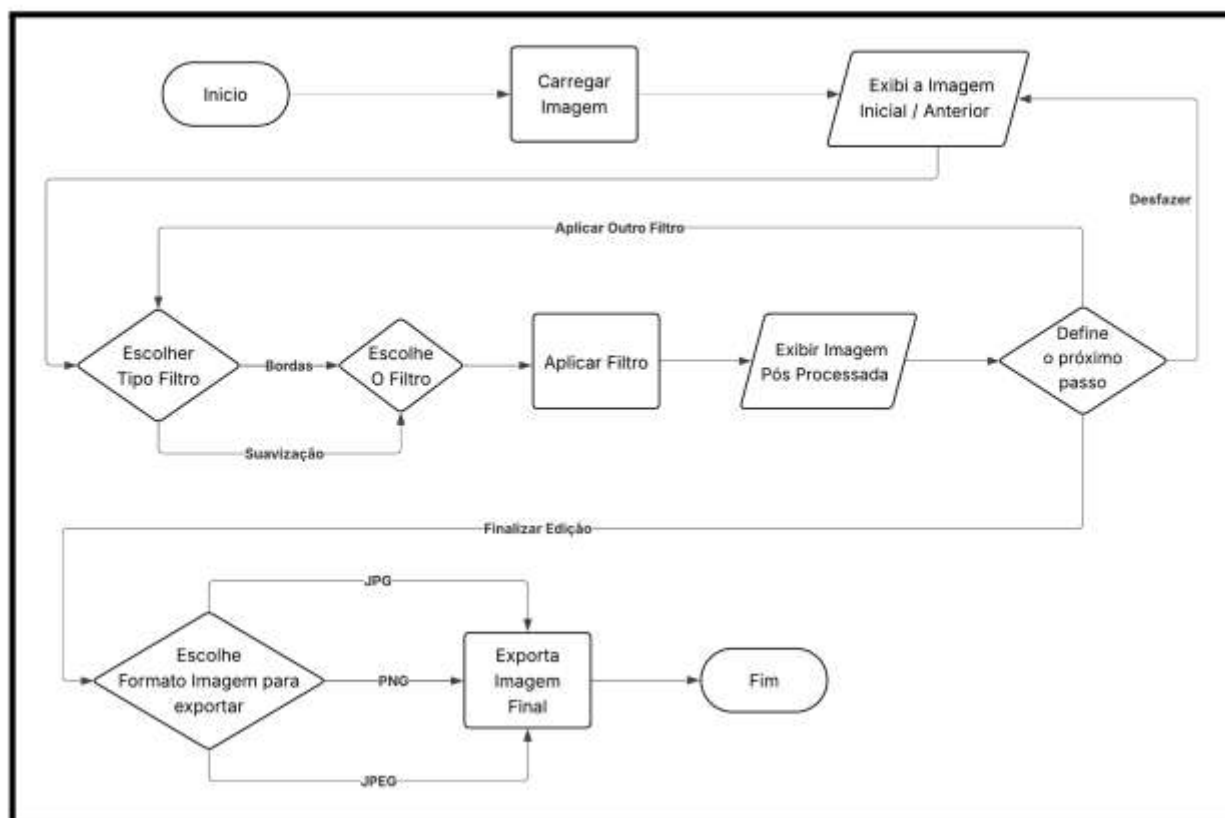


Figura 11 – Estrutura de Fluxo de Processamento



(a) Imagem Original.



(b) Filtro nível 1.



(c) Filtro nível 2.



(d) Filtro nível 3.

Figura 12 – Filtro de Sobel.



(a) Imagem Original.



(b) Filtro nível 1.



(c) Filtro nível 2.



(d) Filtro nível 3.

Figura 13 – Filtro de Roberts.



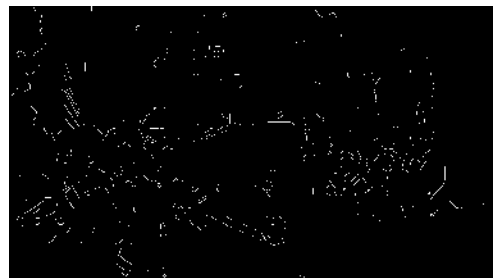
(a) Imagem Original.



(b) Filtro nível 1.



(c) Filtro nível 2.

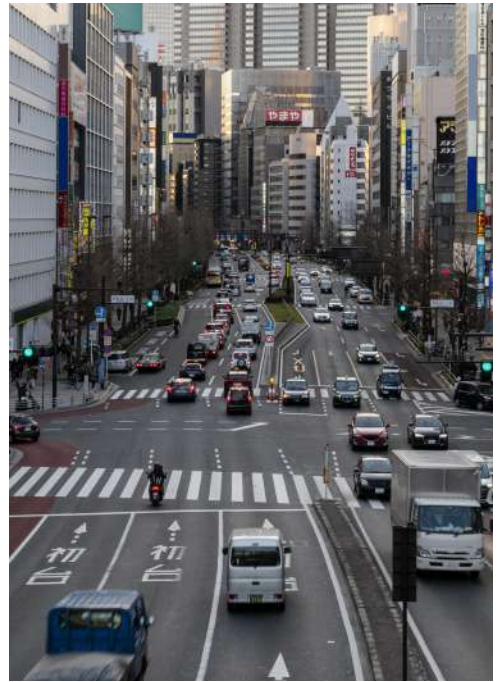


(d) Filtro nível 3.

Figura 14 – Filtro de Canny.



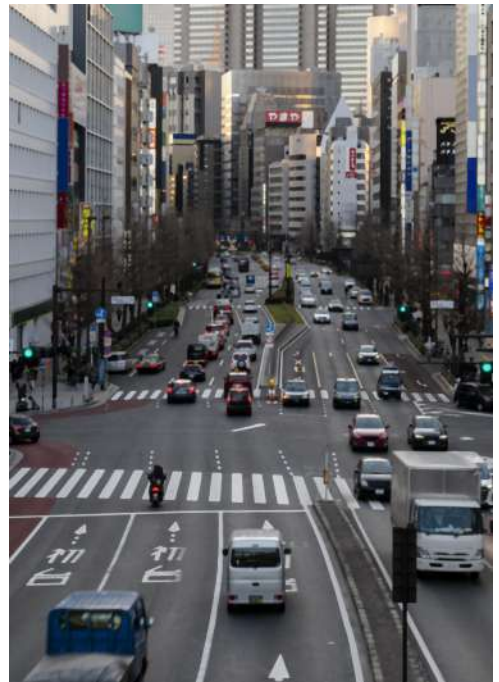
(a) Imagem Original.



(b) Filtro nível 1.



(c) Filtro nível 2.



(d) Filtro nível 3.

Figura 15 – Filtro Gaussiano.



(a) Imagem Original.



(b) Filtro nível 1.



(c) Filtro nível 2.



(d) Filtro nível 3.

Figura 16 – Filtro Bilateral.



(a) Imagem Original.



(b) Filtro nível 1.



(c) Filtro nível 2.



(d) Filtro nível 3.

Figura 17 – Filtro Média.



(a) Imagem Original.



(b) Filtro nível 1.



(c) Filtro nível 2.



(d) Filtro nível 3.

Figura 18 – Filtro Média.



(a) Imagem Original.



(b) Filtro Personalizado.

Figura 19 – Filtro Gaussiano Personalizado.



(a) Imagem Original.



(b) Filtro Personalizado.

Figura 20 – Filtro de Canny Personalizado.